

CLAIMS

What is claimed is:

- 1 1. An object-oriented method for a client application, said method
2 comprising:
3 registering for events occurring during an analysis of a physical layout of a
4 microchip design, including:
5 creating a class to implement a method that is responsive to an event; and
6 registering the event with a server class, wherein the registering includes
7 providing a pointer to the class.
- 1 2. The method of claim 1, wherein the creating includes creating a
2 concrete class that inherits a virtual method of an abstract class.
- 1 3. The method of claim 2, wherein the concrete class executes a callback
2 method.
- 1 4. The method of claim 1, wherein the registering the event includes
2 informing the server class of at least one connectivity event between objects in a
3 physical layout that a client application is to report on.
- 1 5. The method of claim 1, wherein the registering includes executing a
2 register applications programming interface.
- 1 6. The method of claim 1, further including creating a new method to
2 handle a new event while retaining the class.
- 1 7. An object-oriented method for a server class, said method comprising:
2 registering for events occurring during an analysis of a physical layout of a
3 microchip design, including:
4 receiving a request from a class to store an event; and
5 storing the event in a table.

1 8. The method of claim 7, wherein the receiving a request includes
2 receiving the event that the class seeks notification of upon detection.

1 9. The method of claim 7, wherein the storing includes storing the events
2 in a table of callbacks class.

1 10. The method of claim 7, further including instantiating a server class
2 and storing its pointer in a table of callbacks class, the server class being called
3 responsive to the event being detected.

1 11. The method of claim 10, wherein the server class is configured to call a
2 receiver class upon being called in response to the event being detected.

1 12. The method of claim 7, wherein the storing includes implementing an
2 applications programming interface.

1 13. The method of claim 7, further including erasing the event.

1 14. The method of claim 13, wherein the erasing includes implementing an
2 applications programming interface.

1 15. An object-oriented method for a client application, said method
2 comprising:
3 implementing a callback in response to an event in a physical layout of a
4 microchip design, including:
5 receiving a call from a server class to implement a callback;
6 receiving information corresponding to an event in the call; and
7 executing the callback in a receiver class that is decoupled from the server
8 class.

1 16. The method of claim 15, wherein the information includes coordinate
2 information about the objects that correspond to the event.

- 1 17. The method of claim 15, wherein the executing the callback includes
2 executing a method that reports on a detected event.
- 1 18. The method of claim 15, wherein the method includes an applications
2 programming interface.
- 1 19. The method of claim 15, wherein the executing the callback includes
2 reporting events occurring in a physical layout.
- 1 20. The method of claim 19, wherein the reporting events includes at least
2 one of reporting two rectangles touching, two rectangles overlapping, the first
3 encounter of a rectangle, and the encounter of the end of a rectangle.
- 1 21. The method of claim 15, wherein the executing the callback includes
2 executing a method in a concrete receiver class.
- 1 22. The method of claim 15, wherein the executing the callback includes
2 executing a method in an abstract receiver class.
- 1 23. An object-oriented method for a server engine class, said method
2 comprising:
3 initiating a callback in response to an event occurring in a physical layout of a
4 microchip design, including:
5 receiving an indication of the event; and
6 searching a table for an event identifier corresponding to the event and a
7 pointer to an instantiated server class that will make a call to a responsible receiver
8 class.
- 1 24. The method of claim 23, wherein the instantiated server class is an
2 instantiation of a server engine class.
- 1 25. The method of claim 23, further including passing information from
2 the table about the type of event and the objects that caused the event.

1 26. An object-oriented callback system, comprising:
2 a client class configured to receive information corresponding to an event in an
3 artwork of a microchip design, said client class configured to use the information to
4 report on the event.

1 27. The system of claim 26, wherein the client class is configured to
2 register the event with a server class, wherein the registering includes providing a
3 pointer to the client class.

1 28. The system of claim 26, wherein the client class is configured to
2 receive a call from a server class to execute an applications programming interface
3 that corresponds to a callback method.

1 29. The system of claim 26, wherein the client class is decoupled from a
2 server class, said server class configured to provide a scan traversal of the artwork.

1 30. The system of claim 26, wherein the client class includes an abstract
2 receiver class and a concrete receiver class.

1 31. The system of claim 30, wherein the concrete receiver class executes
2 an applications programming interface corresponding to a callback method.

1 32. The system of claim 26, further including a server class decoupled
2 from the client class, said server class configured to detect events in the artwork.

1 33. The system of claim 32, wherein the client class and the server class
2 are configured as a plurality of modules.

1 34. The system of claim 32, wherein the client class and the server class
2 are configured as a single module.

1 35. The system of claim 32, wherein the client class and the server class
2 are configured in software in cooperation with a processor.

1 36. The system of claim 32, wherein the client class and the server class
2 are configured in a digital signal processor.

1 37. The system of claim 32, wherein the client class and the server class
2 are embodied in a computer readable medium.

1 38. An object-oriented callback system, comprising:
2 means for creating a class to implement a method that is responsive to an
3 event;
4 means for detecting the event in artwork of a microchip design;
5 means for passing information about the type of event and information about
6 objects that caused the event to occur; and
7 means for reporting the event.

1 39. A computer readable medium having a computer program comprising
2 an object-oriented method for a client application, comprising:
3 logic configured to register for events occurring during an analysis of a
4 physical layout of a microchip design, including:
5 logic configured to create a class to implement a method that is responsive to
6 an event; and
7 logic configured to register the event with a server class, wherein the
8 registering includes providing a pointer to the class.

1 40. A computer readable medium having a computer program comprising
2 an object-oriented method for a server class, comprising:
3 logic configured to register for events occurring during an analysis of a physical
4 layout of a microchip design, including:
5 logic configured to receive a request from a class to store an event; and
6 logic configured to store the event in a table.

1 41. A computer readable medium having a computer program comprising
 2 an object-oriented method for a client application, comprising:
 3 logic configured to implement a callback in response to an event in a physical
 4 layout of a microchip design, including:
 5 logic configured to receive a call from a server class to implement a callback;
 6 logic configured to receive information corresponding to an event in the call;
 7 and
 8 logic configured to execute the callback in a receiver class that is decoupled
 9 from the server class.

1 42. A computer readable medium having a computer program comprising
 2 an object-oriented method for a server class, comprising:
 3 logic configured to initiate a callback in response to an event occurring in a
 4 physical layout of a microchip design, including:
 5 logic configured to receive an indication of the event; and
 6 logic configured to search a table for an event identifier corresponding to the
 7 event and a pointer to an instantiated server class that will make a call to a responsible
 8 receiver class.